REMARKS

Claims 1-20 are pending in the above-captioned patent application. Claims 1, 10, 16 and 19 are independent claims.

The examiner uses Dyer to reject claim 1 as having been anticipated.

For a reference to anticipate a claim, each element and limitation of the claim must be found in the reference. Hoover Group, Inc. v. Custom Metalcraft, Inc., 66 F.3d 299, 302 (Fed. Cir. 1995). Claim 1, as amended, recites "cause an executing instruction stream to branch to an instruction at an address specified in the instruction if a state indicating a current microengine processing state of a specified state name is a specified value." Dyer fails to describe or suggest the quoted claim feature.

The examiner argues that in applicant's response filed on January 3, 2005:

> **(A)pplicant failed to explain why Dyer's branch mark sequencer instruction format disclosed no state and/or specified state name. The examiner would like to point out Dyer taught a branch instruction that caused an execution of instruction stream to branch to an instruction at an address [801] specified in a state of a specified name (e.g., branch type) was a specified value (see branch type – 0,1 col. 11, lines 8-10, see also the state of additional specified name in Table 1, see fig. 8 for branch instruction format, see also the citations and discussions set forth in the last office action 12/14/04). (Office Action dated March 18, 2005, p. 3-4)**

Dyer discloses an instruction set for a timing mark sequencer having a format described with reference to FIG. 8:

> **(E)ach instruction 800 (FIG. 8) has nine fields: a branch address 801 field, a branch type field 802, a servo timing mark field 803, a sync found clock field 804, a sync found branch field 805, a counter function field 806, a window counter load/count field 807, a space counter load/count field 808, and a count value field 809. Field 801 is stored in region 704 of memory 241. Fields 802 to 808 are stored in region 705 of memory 241 and field 809 is stored in region 706 of memory 241. (Col. 10, lines 56-65)**

No state indicating a current microengine processing state of a specified state name is found in Dyer and therefore Dyer cannot anticipate claim 1. See Hoover Group, Inc. v. Custom Metalcraft, Inc., id.

Table 1 in Dyer presents each of the data fields 801 to 809 instruction 800:

TABLE 1

TIMING MARK SEQUENCER INSTRUCTION FORMAT

Bit (s)	Name	Description/Function
19:16	Branch Address	A branch address which is taken if the branch condition specified in the Branch Type field is true.
15	Branch Type	0 -- Branch on presence of bit HRBIT 1 -- Unconditional Branch, always branch on next clock cycle.
14	Servo Timing Mark	This bit drives the output signal of timing mark sequencer 240. This bit is asserted to generate the active servo timing mark signal STM.
13	Sync Found Clock	If this bit is set and the branch condition specified in the Branch Type field is true, a logic one is clocked into sync found flip-flop 606.
12	Sync Found Branch	If this bit is set and the branch flip-flop 606 has a logic one output signal and Counter Function is true, branch to Branch Address.
11:10	Counter Function	These bits determine how branch and fetch unit 601 uses space counter 604 and window counter 605. 0x -- Ignore counters 10 -- Remain at the current address until the branch condition is true or the value of window counter 605 is zero. 11 -- Remain at the current address until the branch condition is true or the value of space counter 604 is zero.
09	Window Counter Load/Count	1 -- Count Value field is loaded into window counter 605. 0 -- Window counter is decremented by each rising clock edge of decode clock DCLK.
08	Space Counter Load/Count	1 -- Count Value field is loaded into space counter 604. 0 -- Space counter 604 is decremented by each rising clock edge of decode clock DCLK.
07:00	Count Value	Value loaded into either or both of

> **space counter 604 and window counter 605 when their Load/Count bit is set.**

Dyer's "branch type" is clearly very different from applicant's claimed feature. No state indicating a current microengine processing state of a specified state name is found in Dyer and therefore Dyer cannot anticipate claim 1. See Hoover Group, Inc. v. Custom Metalcraft, Inc., id.

As described in applicant's original detailed description with reference to FIG. 5:

> **Referring to FIG. 5, the microengines support various branch instructions such as those that branch on condition codes. In addition, the microengines also support branch instructions that branch on a processor state.**
>
> **A format as shown in FIG. 5 uses br_mask field is used to specify branch. For branch mask = 15, the extended field is used to specify the various signal and state signals as listed below.**
>
> **BR_INP_STATE**
>
> **The BR_INP_STATE instruction branches if a state of a specified state name is set to 1. A state is set to 1 or 0 by a microengine in the processor 10 and indicates the current processing state. The state of the state name is available to all microengines. The instruction can have the following format br_inp_state[state_name, label#], optional_token**

Accordingly, claim 1 is not anticipated by Dyer.

The examiner uses Hasegawa to reject claims 1-20 as having been anticipated.

Claims 1, 10, 16 and 19, as amended, recite "cause an executing instruction stream to branch to an instruction at an address specified in the instruction if a state indicating a current microengine processing state of a specified state name is a specified value," or similar language. Hasegawa neither describes nor suggests this quoted feature.

The examiner argues:

> **Applicant did not explain why Hasegawa's Z and C flags are different from applicant's claimed features, and did not explain why Hasegawa's Z and C flags encoded in the opcode could not be used as the specified state names.**
>
> **Hasegawa disclosed the state of a specified state name (see the value specified in branch instruction field, see also fig. 5, and fig. 10, see implicit value Z and C encoded in Table 1, col. 11, lines 41-52, see also the Z and C flags set and reset in col. 11, lines 14-40, lines 54-67, , see also the citations and discussions set forth in the last office action 12/14/04). (Office Action dated March 18, 2005, p. 3)**

Nothing in Hasegawa describes, suggests, or even mentions, state and/or specified state name. Hasegawa discloses:

The executing section 11 updates the value of the condition code 61 based on the execution result 109. The condition code 61 is composed of four bits: Z (1 bit), N (1 bit), V (1 bit) and C (1 bit), for example. Z denotes a zero flag, N denotes a negative flag, V denotes an overflow flag and C denotes a carry flag. Each of these flags has a value, for example, of 0 or 1. The values of these flags are updated by the executing section 11. (Col. 11, lines 25-32)

The above indicates that the Z bit condition code denotes a zero flag and the C flag condition code denotes a carry flag.

A predictive branch instruction has a region 21 for storing an opecode therein, a region 22 for specifying a branch target address, and a region 23 for storing the number of at least one instruction which is to be executed in succession after the predictive branch instruction is given before the control flow is changed, i.e., a branch point.

An "opecode" is a code for identifying a kind of a given instruction. An opecode is generally composed of a plurality of bits.

A branch target address is specified, for example, in a direct addressing mode or in an indirect addressing mode. (Col. 6, lines 1-11)

Hawesaga's disclosed opecode cannot explicitly or implicitly be considered the same as applicant's state indicating a current microengine processing state of a specified state name at least because Hawesaga's opecode is a code for identifying a kind of given instruction.

Further, as applicant set out in a previous response dated January 3, 2005, Hasegawa hard codes opecodes referencing branch conditions in a region 21 as set out in Table 1 in col. 11, while applicant's feature claim dynamically causes an executing instruction stream to branch to an instruction at an address specified in the instruction if a state indicating a current microengine processing state of a specified state name is a specified value.

Accordingly, claims 1, 10, 16 and 19 are not anticipated by Hasegawa.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue or comment does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Applicant asks that all claims be examined in view of the amendment to the claims.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: June 14, 2005

Kenneth F. Kozik
Reg. No. 36,572

ATTORNEYS FOR INTEL
Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21107501.doc